

09/528, 466

FTP Transfer Log
≤ 29 November 1998

DOCUMENT H

- (1) 98.11.09 09:30 B C: \incoming\crib_database\cgi\derek\admin.cgi <- matthew /export/home/howie\crib_database\cgi\derek admin.cgi
- (2) 98.11.09 09:30 B C: \incoming\crib_database\cgi\derek\admin_del.cgi <- matthew /export/home/howie\crib_database\cgi\derek admin_del.cgi
- (3) 98.11.09 09:30 B C: \incoming\crib_database\cgi\derek\admin_edit.cgi <- matthew /export/home/howie\crib_database\cgi\derek admin_edit.cgi
- (4) 98.11.09 09:30 B C: \incoming\crib_database\cgi\derek\admin_make.cgi <- matthew /export/home/howie\crib_database\cgi\derek admin_make.cgi
- (5) 98.11.09 09:30 B C: \incoming\crib_database\cgi\derek\admin_modify.cgi <- matthew /export/home/howie\crib_database\cgi\derek admin_modify.cgi
- (6) 98.11.09 09:30 B C: \incoming\crib_database\cgi\derek\admin_result.cgi <- matthew /export/home/howie\crib_database\cgi\derek admin_result.cgi
- (7) 98.11.09 09:30 B C: \incoming\crib_database\cgi\derek\admin_toggle.cgi <- matthew /export/home/howie\crib_database\cgi\derek admin_toggle.cgi
- (8) 98.11.09 09:30 B C: \incoming\crib_database\cgi\derek\liberty_shared.cgi <- matthew /export/home/howie\crib_database\cgi\derek liberty_shared.cgi
- (9) 98.11.09 09:30 B C: \incoming\crib_database\cgi\derek\liberty_view.html <- matthew /export/home/howie\crib_database\cgi\derek liberty_view.html
- (10) 98.11.09 09:30 B C: \incoming\crib_database\cgi\derek\read_query.cgi <- matthew /export/home/howie\crib_database\cgi\derek read_query.cgi
- (11) 98.11.09 09:30 B C: \incoming\crib_database\cgi\derek\read_result.cgi <- matthew /export/home/howie\crib_database\cgi\derek read_result.cgi
- (12) 98.11.09 09:30 B C: \incoming\crib_database\cgi\derek\vw_result.html <- matthew /export/home/howie\crib_database\cgi\derek vw_result.html

(2)

```
#!/usr/bin/perl
require ('liberty_shared.cgi');

&HTML;
&GETSET;
&GETFORMDATA;
&GETDATABASE;

$DelLin = $FORM{id};

#PartNo

open (DB, ">$db") ;

for ($i=0;$i<=$FilesInDatabase;$i++) {
if ($PartNo[$i] ne $DelLin) {
    print DB
"$State[$i]\t$Establishment[$i]\t$Address[$i]\t$City[$i]\t$Phone[$i]\t$WebLoc[$i]
]\t$PartNo[$i]\t$Group[$i]\t$Password[$i]\t$Account[$i]\t$Email[$i]\t$Active[$i]
\t$ActWWW\n";
}
}
close (<DB>);

print "Account $DelLin Removed.<P>";
print "<A HREF=\"admin.cgi\">Click Here to Return to Main Admin Screen</A><BR>";
print "Do not use your browser back button.";
```

(3)

```
#!/usr/bin/perl
require ('liberty_shared.cgi');

&HTML;
&GETSET;
&GETFORMDATA;
&GETDATABASE;

$DelLin = $FORM{id};

#PartNo

open (DB, ">$db") ;

for ($i=0;$i<=$FilesInDatabase;$i++) {
if ($PartNo[$i] ne $DelLin) {
    print DB
"$State[$i]\t$Establishment[$i]\t$Address[$i]\t$City[$i]\t$Phone[$i]\t$WebLoc[$i]
]\t$PartNo[$i]\t$Group[$i]\t$Password[$i]\t$Account[$i]\t$Email[$i]\t$Active[$i]
\t$ActWWW\n";
}
}
close (<DB>);

print "Account $DelLin Removed.<P>";
print "<A HREF=\"admin.cgi\">Click Here to Return to Main Admin Screen</A><BR>";
print "Do not use your browser back button.";
```

(4)

```
#!/usr/bin/perl
require ('liberty_shared.cgi');

&HTML;
&GETSET;
&GETFORMDATA;
&GETDATABASE;

$UID = $FORM{add};

#PartNo

open (DB, ">>$db");

print DB "$FORM{stat}\t";
print DB "$FORM{estab}\t";
print DB "$FORM{addr}\t";
print DB "$FORM{city}\t";
print DB "$FORM{phon}\t";
print DB "$FORM{wweb}\t";
print DB "$UID\t";
print DB "$FORM{grup}\t";
print DB "$FORM{pass}\t";
print DB "$FORM{acct}\t";
print DB "$FORM{mail}\t";
print DB "$FORM{actv}\t";
print DB "$FORM{actwww}\n";

close (<DB>);

print "New account recorded.<P>";
print "<A HREF=\"admin.cgi\">Click Here to Return to Main Admin Screen</A><BR>";
print "Do not use your browser back button.";
```

(5)

```
#!/usr/bin/perl
require ('liberty_shared.cgi');

&HTML;
&GETSET;
&GETFORMDATA;
&GETDATABASE;

$RepLin = $FORM{modify};

#PartNo

open (DB, ">$db");

for ($i=0;$i<=$FilesInDatabase;$i++) {
if ($PartNo[$i] eq $RepLin) {
    print DB "$FORM{stat}\t";
    print DB "$FORM{estab}\t";
    print DB "$FORM{addr}\t";
    print DB "$FORM{city}\t";
    print DB "$FORM{phon}\t";
    print DB "$FORM{wweb}\t";
    print DB "$RepLin\t";
    print DB "$FORM{grup}\t";
    print DB "$FORM{pass}\t";
    print DB "$FORM{acct}\t";
    print DB "$FORM{mail}\t";
    print DB "$FORM{actv}\t";
    print DB "$FORM{actwww}\t";
}
else {
    print DB
"$State[$i]\t$Establishment[$i]\t$Address[$i]\t$City[$i]\t$Phone[$i]\t$WebLoc[$i]
\t$PartNo[$i]\t$Group[$i]\t$Password[$i]\t$Account[$i]\t$Email[$i]\t$Active[$i]
\t$ActWWW[$i]\n";
}
}
close (<DB>);

print "Changes recorded.<P>";
print "<A HREF=\"admin.cgi\">Click Here to Return to Main Admin Screen</A><BR>";
print "Do not use your browser back button.";
```

(6)



```

EOF if ($Action eq "group") { &ADMINSELECTGROUP; } elsif ($Action eq "region")
{ &ADMINSELECTREGION; } elsif ($Action eq "account")
{ &ADMINSELECTACCOUNT; } elsif ($Action eq "create")
{ &ADMINCREATEACCOUNT; } else { &ADMINSELECTALL; } sub
ADMINSELECTGROUP { $SelNum = 0; for ($z=0;$z<=$FilesInDatabase;$z++) { if
($rGroup eq $Group[$z]) { if ($rSort eq "stat") { $SELECT[$SelNum] =
"$State[$z]\t$Establishment[$z]\t$Address[$z]\t$City[$z]\t$Phone[$z]\t$WebLoc[$z]\t$Group[$z]\t$Password[$z]\t$Account[$z]\t$Email[$z]\t$Active[$z]\t$PartNo[$z]\t$Act WWW[$z]"; } elsif ($rSort eq "esta") { $SELECT[$SelNum] =
"$Establishment[$z]\t$Address[$z]\t$City[$z]\t$State[$z]\t$Phone[$z]\t$WebLoc[$z]\t$Group[$z]\t$Password[$z]\t$Account[$z]\t$Email[$z]\t$Active[$z]\t$PartNo[$z]\t$Act WWW[$z]"; } elsif ($rSort eq "addr") { $SELECT[$SelNum] =
"$Address[$z]\t$Establishment[$z]\t$City[$z]\t$State[$z]\t$Phone[$z]\t$WebLoc[$z]\t$Group[$z]\t$Password[$z]\t$Account[$z]\t$Email[$z]\t$Active[$z]\t$PartNo[$z]\t$Act WWW[$z]"; } elsif ($rSort eq "city") { $SELECT[$SelNum] =
"$City[$z]\t$Establishment[$z]\t$Address[$z]\t$State[$z]\t$Phone[$z]\t$WebLoc[$z]\t$Group[$z]\t$Password[$z]\t$Account[$z]\t$Email[$z]\t$Active[$z]\t$PartNo[$z]\t$Act WWW[$z]"; } elsif ($rSort eq "grou") { $SELECT[$SelNum] =
"$Group[$z]\t$Establishment[$z]\t$Address[$z]\t$City[$z]\t$State[$z]\t$Phone[$z]\t$WebLoc[$z]\t$Password[$z]\t$Account[$z]\t$Email[$z]\t$Active[$z]\t$PartNo[$z]\t$Act WWW[$z]"; } $SelNum++; } } $SelNum = $SelNum - 1; @SELECT = sort(@SELECT);
print ""; print "[MARK ENTIRE GROUP ON]"; print "[MARK ENTIRE GROUP OFF]"

\n"; print "

\n"; for ($z=0;$z<=$SelNum;$z++) { if ($rSort eq "stat")
{ ($sta,$est,$add,$cty,$pho,$www,$grp,$pwd,$acc,$ema,$act,$num,$actww) = split (/t/, $SELECT[$z]); } elsif ($rSort eq "esta")
{ ($est,$add,$cty,$sta,$pho,$www,$grp,$pwd,$acc,$ema,$act,$num,$actww) = split (/t/, $SELECT[$z]); } elsif ($rSort eq "addr")
{ ($add,$est,$cty,$sta,$pho,$www,$grp,$pwd,$acc,$ema,$act,$num,$actww) = split (/t/, $SELECT[$z]); } elsif ($rSort eq "city")
{ ($cty,$est,$add,$sta,$pho,$www,$grp,$pwd,$acc,$ema,$act,$num,$actww) = split (/t/, $SELECT[$z]); } elsif ($rSort eq "grou")
{ ($grp,$est,$add,$cty,$sta,$pho,$www,$pwd,$acc,$ema,$act,$num,$actww) = split (/t/, $SELECT[$z]); } elsif ($rSort eq "acco")
{ ($acc,$est,$add,$cty,$sta,$pho,$www,$grp,$pwd,$ema,$act,$num,$actww) = split (/t/, $SELECT[$z]); } $act =~ tr/a-z/A-Z/; if (($rInactive eq
"hide") and ($act eq "OFF")) { print "\n"; } else { print <$est [DELETE]
$add
$cty, $sta
$pho
$www ($actwww)
}

```

```

Account Name: $acc [$act]
Contact Mail: $ema
Grouped with: $grp

EOF } } print "

"; } sub ADMINSELECTREGION { $SelNum = 0; for
($z=0;$z<=$FilesInDatabase;$z++) { if ($rRegion eq $State[$z]) { if ($rSort eq "stat")
{ $SELECT[$SelNum] =
"$State[$z]\t$Establishment[$z]\t$Address[$z]\t$City[$z]\t$Phone[$z]\t$WebLoc[$z]\t$Group[$z]\t$Password[$z]\t$Account[$z]\t$Email[$z]\t$Active[$z]\t$PartNo[$z]\t$Act WWW[$z]"; } elsif ($rSort eq "esta") { $SELECT[$SelNum] =
"$Establishment[$z]\t$Address[$z]\t$City[$z]\t$State[$z]\t$Phone[$z]\t$WebLoc[$z]\t$Group[$z]\t$Password[$z]\t$Account[$z]\t$Email[$z]\t$Active[$z]\t$PartNo[$z]\t$Act WWW[$z]"; } elsif ($rSort eq "addr") { $SELECT[$SelNum] =
"$Address[$z]\t$Establishment[$z]\t$City[$z]\t$State[$z]\t$Phone[$z]\t$WebLoc[$z]\t$Group[$z]\t$Password[$z]\t$Account[$z]\t$Email[$z]\t$Active[$z]\t$PartNo[$z]\t$Act WWW[$z]"; } elsif ($rSort eq "city") { $SELECT[$SelNum] =
"$City[$z]\t$Establishment[$z]\t$Address[$z]\t$State[$z]\t$Phone[$z]\t$WebLoc[$z]\t$Group[$z]\t$Password[$z]\t$Account[$z]\t$Email[$z]\t$Active[$z]\t$PartNo[$z]\t$Act WWW[$z]"; } elsif ($rSort eq "grou") { $SELECT[$SelNum] =
"$Group[$z]\t$Establishment[$z]\t$Address[$z]\t$City[$z]\t$State[$z]\t$Phone[$z]\t$WebLoc[$z]\t$Password[$z]\t$Account[$z]\t$Email[$z]\t$Active[$z]\t$PartNo[$z]\t$Act WWW[$z]"; } $SelNum++; } } $SelNum = $SelNum - 1; @SELECT = sort(@SELECT);
print "
\n"; for ($z=0;$z<=$SelNum;$z++) { if ($rSort eq "stat")
{ ($sta,$est,$add,$cty,$pho,$www,$grp,$pwd,$acc,$ema,$act,$num,$actwww) = split
(/\t/, $SELECT[$z]); } elsif ($rSort eq "esta")
{ ($est,$add,$cty,$sta,$pho,$www,$grp,$pwd,$acc,$ema,$act,$num,$actwww) = split
(/\t/, $SELECT[$z]); } elsif ($rSort eq "addr")
{ ($add,$est,$cty,$sta,$pho,$www,$grp,$pwd,$acc,$ema,$act,$num,$actwww) = split
(/\t/, $SELECT[$z]); } elsif ($rSort eq "city")
{ ($cty,$est,$add,$sta,$pho,$www,$grp,$pwd,$acc,$ema,$act,$num,$actwww) = split
(/\t/, $SELECT[$z]); } elsif ($rSort eq "grou")
{ ($grp,$est,$add,$cty,$sta,$pho,$www,$pwd,$acc,$ema,$act,$num,$actwww) = split
(/\t/, $SELECT[$z]); } elsif ($rSort eq "acco")
{ ($acc,$est,$add,$cty,$sta,$pho,$www,$grp,$pwd,$ema,$act,$num,$actwww) = split
(/\t/, $SELECT[$z]); } $act =~ tr/a-z/A-Z/; if (($rInactive eq "hide") and ($act eq "OFF"))
{ print "\n"; } else { print <$est [DELETE]
$add
$cty, $sta
$pho
$www ($actwww)
Account Name: $acc [$act]
Contact Mail: $ema
Grouped with: $grp

```

```

EOF } } print "

"; } sub ADMINSELECTACCOUNT { $SelEntry = "None"; for
($z=0;$z<=$FilesInDatabase;$z++) { if ($rAccountName eq $Account[$z])
{ $SelEntry=$i; last; } } if ($SelEntry eq "None") { print "No account found."; } else
{ print <$Establishment[$SelEntry] [DELETE]
$Address[$SelEntry]
$City[$SelEntry], $State[$SelEntry]
$Phone[$SelEntry]
$WebLoc[$SelEntry] ($ActWWW[$SelEntry])
Account Name: $Account[$SelEntry] [$Active[$SelEntry]]
Contact Mail: $Email[$SelEntry]
Grouped with: $Group[$SelEntry]

EOF } } sub ADMINSELECTALL { $SortNum = $FilesInDatabase; for
($z=0;$z<=$FilesInDatabase;$z++) { if ($rSort eq "stat") { $SELECT[$z] =
"$State[$z]\t$Establishment[$z]\t$Address[$z]\t$City[$z]\t$Phone[$z]\t$WebLoc[$z]\t$Group[$z]\t$Password[$z]\t$Account[$z]\t$Email[$z]\t$Active[$z]\t$PartNo[$z]\t$Act WWW[$z]"; } elsif ($rSort eq "esta") { $SELECT[$z] =
"$Establishment[$z]\t$Address[$z]\t$City[$z]\t$State[$z]\t$Phone[$z]\t$WebLoc[$z]\t$Group[$z]\t$Password[$z]\t$Account[$z]\t$Email[$z]\t$Active[$z]\t$PartNo[$z]\t$Act WWW[$z]"; } elsif ($rSort eq "addr") { $SELECT[$z] =
"$Address[$z]\t$Establishment[$z]\t$City[$z]\t$State[$z]\t$Phone[$z]\t$WebLoc[$z]\t$Group[$z]\t$Password[$z]\t$Account[$z]\t$Email[$z]\t$Active[$z]\t$PartNo[$z]\t$Act WWW[$z]"; } elsif ($rSort eq "city") { $SELECT[$z] =
"$City[$z]\t$Establishment[$z]\t$Address[$z]\t$State[$z]\t$Phone[$z]\t$WebLoc[$z]\t$Group[$z]\t$Password[$z]\t$Account[$z]\t$Email[$z]\t$Active[$z]\t$PartNo[$z]\t$Act WWW[$z]"; } elsif ($rSort eq "grou") { $SELECT[$z] =
"$Group[$z]\t$Establishment[$z]\t$Address[$z]\t$City[$z]\t$State[$z]\t$Phone[$z]\t$WebLoc[$z]\t$Password[$z]\t$Account[$z]\t$Email[$z]\t$Active[$z]\t$PartNo[$z]\t$Act WWW[$z]"; } } @SELECT = sort(@SELECT); print "

\n"; for ($z=0;$z<=$SortNum;$z++) { if ($rSort eq "stat")
{ ($sta,$est,$add,$cty,$pho,$www,$grp,$pwd,$acc,$ema,$act,$num,$actw
ww) = split (/t/, $SELECT[$z]); } elsif ($rSort eq "esta")
{ ($est,$add,$cty,$sta,$pho,$www,$grp,$pwd,$acc,$ema,$act,$num,$actw
ww) = split (/t/, $SELECT[$z]); } elsif ($rSort eq "addr")
{ ($add,$est,$cty,$sta,$pho,$www,$grp,$pwd,$acc,$ema,$act,$num,$actw
ww) = split (/t/, $SELECT[$z]); } elsif ($rSort eq "city")
{ ($cty,$est,$add,$sta,$pho,$www,$grp,$pwd,$acc,$ema,$act,$num,$actw
ww) = split (/t/, $SELECT[$z]); } elsif ($rSort eq "grou")
{ ($grp,$est,$add,$cty,$sta,$pho,$www,$pwd,$acc,$ema,$act,$num,$actw
ww) = split (/t/, $SELECT[$z]); } elsif ($rSort eq "acco")
{ ($acc,$est,$add,$cty,$sta,$pho,$www,$grp,$pwd,$ema,$act,$num,$actw
ww) = split (/t/, $SELECT[$z]); } $act =~ tr/a-z/A-Z/; if (($rInactive eq
"hide") and ($act eq "OFF")) { print "\n"; } else { print <$est [DELETE]
$add
}

```

```

$cty, $sta
$pho
$www ($actwww)
Account Name: $acc [$act]
Contact Mail: $ema
Grouped with: $grp

EOF } } print "

"; } sub ADMINCREATEACCOUNT { $UID=0; for ($i=0;$i<=$FilesInDatabase;$i++)
{ $cpt = $PartNo[$i] + 0; if ($cpt > $UID) { $UID = $cpt; } } $UID++; if (length ($UID)
eq 1) { $UID = "0000$UID"; } elsif (length ($UID) eq 2) { $UID = "000$UID"; } elsif
(length ($UID) eq 3) { $UID = "00$UID"; } elsif (length ($UID) eq 4) { $UID =
"0$UID"; } print <

```

Establishment	<input type="text"/>
Group	<input type="text"/>
Address	<input type="text"/>
City	<input type="text"/>
State	<input type="text"/>
Phone	<input type="text"/>
Website	<input type="text"/>
Web Link	<input checked="" type="radio"/> ON. <input type="radio"/> OFF.
Password	<input type="text"/>
Account Name	<input type="text"/>
Email	<input type="text"/>
Account Status	<input checked="" type="radio"/> ON. <input type="radio"/> OFF.
<input type="button" value="Submit"/> <input type="button" value="Reset"/>	

EOF }

⑦

```
#!/usr/bin/perl
require ('liberty_shared.cgi');

&HTML;
&GETSET;
&GETFORMDATA;
&GETDATABASE;

$ToggleGroup = $FORM{group};
$ToggleToggle= $FORM{toggle};

open (DB, ">$db") ;

for ($i=0;$i<=$FilesInDatabase;$i++) {
if ($Group[$i] ne $ToggleGroup) { print DB
"$State[$i]\t$Establishment[$i]\t$Address[$i]\t$City[$i]\t$Phone[$i]\t$WebLoc[$i
]\t$PartNo[$i]\t$Group[$i]\t$Password[$i]\t$Account[$i]\t$Email[$i]\t$Active[$i]
\t$ActWWW[$i]\n"; }

else { print DB
"$State[$i]\t$Establishment[$i]\t$Address[$i]\t$City[$i]\t$Phone[$i]\t$WebLoc[$i
]\t$PartNo[$i]\t$Group[$i]\t$Password[$i]\t$Account[$i]\t$Email[$i]\t$ToggleTogg
le\t$ActWWW[$i]\n"; }
}

close (<DB>);

print "$ToggleGroup marked $ToggleToggle.<P>";
print "<A HREF=\"admin.cgi\">Click Here to Return to Main Admin Screen</A><BR>";
print "Do not use your browser back button.";
```

(8)

```

1; sub HTML { print "Content-type: text/html\n\n"; } sub GETSET { $db =
'/var/data/home/derek/www/retailstore.db'; $img =
'http://www.creek.net/~derek/misc/devbin/img'; $URL =
'http://www.creek.net/~derek/cgi-bin'; } sub GETFORMDATA { if
($ENV{'REQUEST_METHOD'} eq "GET") { $formData =
$ENV{'QUERY_STRING'}; } elsif ($ENV{'REQUEST_METHOD'} eq "POST")
{ read(STDIN, $formData, $ENV{'CONTENT_LENGTH'}); } @pairs = split(/&/,
$formData); foreach $pair (@pairs) { ($name, $value) = split(/=/, $pair); $value =~ tr/+/ /;
$value =~ s/~/!/ ~!/g; $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
$FORM{$name} = $value; } $SeekBy=$FORM{region}; $SortBy=$FORM{sort}; } sub
GETDATABASE { $cnt = 0; open (DB,"") { chomp; ($State[$cnt], $Establishment[$cnt],
$Address[$cnt], $City[$cnt], $Phone[$cnt], $WebLoc[$cnt], $PartNo[$cnt],
$Group[$cnt], $Password[$cnt], $Account[$cnt], $Email[$cnt], $Active[$cnt],
$ActWWW[$cnt]) = split(/\t/); $cnt++; } close (); $cnt = $cnt - 1; $FilesInDatabase =
$cnt; } sub GETACTIVEREGIONS { $reg = 0; for ($i=0;$i<=$FilesInDatabase;$i++)
{ $match = "no"; for ($j=0;$j<=$reg;$j++) { if (($State[$i] eq $Regions[$j]) and
($Active[$i] ne "OFF")) { $match = "yes"; } } if (($match eq "no") and ($Active[$i] ne
"OFF")) { $Regions[$reg] = $State[$i]; $reg++; } } $reg = $reg - 1; @Regions = sort
(@Regions); } sub GETACTIVEGROUPS { $reg = 0; for
($i=0;$i<=$FilesInDatabase;$i++) { $match = "no"; for ($j=0;$j<=$reg;$j++) { if
($Group[$i] eq $Groupings[$j]) { $match = "yes"; } } if ($match eq "no")
{ $Groupings[$reg] = $Group[$i]; $reg++; } } $reg = $reg - 1; $GroupNum = $reg;
@Groupings = sort (@Groupings); } sub SORTACTION { $vw = 0; for
($z=0;$z<=$FilesInDatabase;$z++) { if (($State[$z] eq $SeekBy) and ($Active[$z] ne
"OFF")) { $ViewSort[$vw] =
"$WebLoc[$z]\t$Phone[$z]\t$Group[$z]\t$PartNo[$z]\t$ActWWW[$z]"; if ($SortBy eq
"City") { $ViewSort[$vw] = "$City[$z]\t$Establishment[$z]\t$Address[$z]\t".
$ViewSort[$vw]; $vw++; } elsif ($SortBy eq "Address") { $ViewSort[$vw] =
"$Address[$z]\t$Establishment[$z]\t$City[$z]\t". $ViewSort[$vw]; $vw++; } else
{ $ViewSort[$vw] = "$Establishment[$z]\t$Address[$z]\t$City[$z]\t". $ViewSort[$vw];
$vw++; } } } @ViewSort = sort(@ViewSort); $ViewFiles = $vw - 1; print "

```

\n"; print "Establishme nt\n"; print "	Address \n"; print "	City \n"; print "	Phone\n"; for (\$y=0;\$y<=\$ViewFiles;\$y++) { if (\$SortBy eq "City") { (\$cty,\$est,\$adr,\$www,\$tel,\$grp,\$num,\$acw) = split (\t,\$ViewSort[\$y]); } elsif (\$SortBy eq "Address") { (\$adr,\$est,\$cty,\$www,\$tel,\$grp,\$num,\$acw)=split(\t ,\$ViewSort[\$y]); } else { (\$est,\$adr,\$cty,\$www,\$tel,\$grp,\$num,\$acw) = split(\t,\$ViewSort[\$y]); } print "
\$est "; if (\$acw ne "OFF") { print "\$www"; } print "\n"; print "	\$adr\n	\$cty \n	\$tel\n"; } print "

(9)

The page cannot be found

The page you are looking for might have been removed, had its name changed, or is temporarily unavailable.

Please try the following:

- If you typed the page address in the Address bar, make sure that it is spelled correctly.
- Open the www.creek.net home page, and then look for links to the information you want.
- Click the [Back](#) button to try another link.

HTTP 404 - File not found
Internet Information Services

Technical Information (for support personnel)

- More information:
[Microsoft Support](#)

The page cannot be found

The page you are looking for might have been removed, had its name changed, or is temporarily unavailable.

Please try the following:

- If you typed the page address in the Address bar, make sure that it is spelled correctly.
- Open the www.creek.net home page, and then look for links to the information you want.
- Click the Back button to try another link.

HTTP 404 - File not found

Internet Information Services

Technical Information (for support personnel)

- More information:
[Microsoft Support](#)

(10)



RETAIL STORES NEAR YOU -- SELECT A LOCATION BELOW

GEOGRAPHIC REGION SORT METHOD

  

From the above pulldown selectors, please specify a geographic region and a sort method. Then, press the 'Display Stores' button.

11

EOF & SORTACTION; print <

(6) 98.11.09 09:31 B C: ncoming\crib_database\cgi\cgi-lib.pl <-- matthew
/export/home/howie/ib_database/cgi cgi-lib.pl
(7) 98.11.09 09:31 B C: ncoming\crib_database\cgi\js-load.htm <-- matthew
/export/home/howie/ib_database/cgi js-load.htm
(8) 98.11.09 09:31 B C: ncoming\crib_database\cgi\js-load.htm.save <-- matthew
/export/home/howie/ib_database/cgi js-load.htm.save
(9) 98.11.09 09:31 B C: ncoming\crib_database\cgi\js-load1.htm <-- matthew
/export/home/howie/ib_database/cgi js-load1.htm
(10) 98.11.09 09:31 B C: ncoming\crib_database\cgi\js-load1.htm.save <-- matthew
/export/home/howie/ib_database/cgi js-load1.htm.save
(11) 98.11.09 09:31 B C: ncoming\crib_database\cgi\js-load2.htm <-- matthew
/export/home/howie/ib_database/cgi js-load2.htm
(12) 98.11.09 09:31 B C: ncoming\crib_database\cgi\js-load3.htm <-- matthew
/export/home/howie/ib_database/cgi js-load3.htm
(13) 98.11.09 09:31 B C: ncoming\crib_database\cgi\js-load4.htm <-- matthew
/export/home/howie/ib_database/cgi js-load4.htm
(14) 98.11.09 09:31 B C: ncoming\crib_database\cgi\js-loadc.htm <-- matthew
/export/home/howie/ib_database/cgi js-loadc.htm
(15) 98.11.09 09:31 B C: ncoming\crib_database\cgi\js-loadc.htm.sav <-- matthew
/export/home/howie/ib_database/cgi js-loadc.htm.sav
(16) 98.11.09 09:31 B C: ncoming\crib_database\cgi\library-frames.cgi <-- matthew
/export/home/howie/ib_database/cgi library-frames.cgi
(17) 98.11.09 09:31 B C: ncoming\crib_database\cgi\library-ro.cgi <-- matthew
/export/home/howie/ib_database/cgi library-ro.cgi
(18) 98.11.09 09:31 B C: ncoming\crib_database\cgi\library.cgi <-- matthew
/export/home/howie/ib_database/cgi library.cgi
(19) 98.11.09 09:32 B C: ncoming\crib_database\cgi\lmlvdata <-- matthew
/export/home/howie/ib_database/cgi lmlvdata
(20) 98.11.09 09:32 B C: ncoming\crib_database\cgi\lmlvdata.save <-- matthew
/export/home/howie/ib_database/cgi lmlvdata.save
(21) 98.11.09 09:32 B C: ncoming\crib_database\cgi\netmask.cgi <-- matthew
/export/home/howie/ib_database/cgi netmask.cgi
(22) 98.11.09 09:32 B C: ncoming\crib_database\cgi\retailstore.db <-- matthew
/export/home/howie/ib_database/cgi retailstore.db

Ø

\$msg[0]

```
\n"; foreach $i (1 .. $#msg) { print "
```

```
$msg[$i]
```

```
\n"; } print "
```

(2)

LOADING MULTIPLE FRAMES

(4)

Page 1

Page 2

(6)

(7)

Page 3

(7)

(8)

Page 4

LOADING MULTIPLE FRAMES

(9)

Display 1

Display 2

Display 3

Display 4

About This Example

View Source

@PJL SET PAGEPROTECT=AUTO

@PJL ENTER LANGUAGE=PCL

(1)

(12)

```
#!/usr/local/bin/perl
#
# perl script to track MCST Library items

# menu variables
@TYPES=('Doc', 'License', 'CD', 'Disk', 'Tape');
@CLASSES=('Network', 'BU', 'OS', 'Other', 'Cluster');
@LOCS=('Mediabox', 'Shelf_1', 'Shelf_2', 'Shelf_3', 'Shelf_4');
$SORTBY="Establishment";

# program data directories
$$DATADIR="/opt/HTTPS/WWW/staff/library";
$$LOG="$DATADIR/library.log";
$$STAFF="/opt/HTTPS/WWW/staff/staff.txt";
$DATADIR="/www/htdocs/MANHATTAN/howie/cgi/library";
$$DATADIR="/www/htdocs/howie/cgi/library";
$LOG="$DATADIR/library.log";

# get remote user
open(foo, "finger \@${ENV{'REMOTE_HOST'}} |");
while(<foo>){ if(/(^S+).*console.*/) { $RUSER=$1 } }
close foo;

$RUSER=${ENV{'REMOTE_USER'}};
$RHOST=${ENV{'REMOTE_ADDR'}};

#echo REMOTE_HOST = $REMOTE_HOST
#echo REMOTE_ADDR = $REMOTE_ADDR
#echo REMOTE_USER = $REMOTE_USER

# set some default variables
chop($DATE=`/usr/bin/date +%m/%d/%y`);
chop($FDATE=`/usr/bin/date`);

# process QUERY_STRING
#push(@INC, "/opt/HTTPS/WWW/cgi");
push(@INC, "/www/htdocs/MANHATTAN/howie/cgi");
#push(@INC, "/www/htdocs/howie/cgi");
require 'cgi-lib.pl';
&ReadParse(*in);
foreach $key(keys(%in)){
    if($key eq STATE){$STATE=$in{$key}}
    if($key eq ESTABLISHMENT){$ESTABLISHMENT=$in{$key}}
    if($key eq ADDRESS){$ADDRESS=$in{$key}}
    if($key eq PARTNO){$PARTNO=$in{$key}}
    if($key eq CITY){$CITY=$in{$key}}
    if($key eq PHONENUM){$PHONENUM=$in{$key}}
    if($key eq LOC){$LOC=$in{$key}}
    if($key eq SORTBY){$SORTBY=$in{$key}}
    if($key eq SUBM){$SUBM=$in{$key}}
    if($key eq SUBMS){$SUBMS=$in{$key}}
}

# item add
if("$SUBM" eq "Add/Update" || "$SUBM" eq "Edit"){
    &get_lock;
    $SPARTNO="$PARTNO";
```

```

$SDB="$STATE $ESTABLISHMENT $ADDRESS $CITY      $PHONENUM $LOC
      ${PARTNO}";
&open_db;
$db{$SPARTNO}="$SDB";
&write_db;
&rel_lock;
open(logger,>> $LOG");
    print logger "$RUSER@$RHOST - Add $ESTABLISHMENT - $URL - $FDATE\n";
close logger;
# link edit
}elseif("$SUBM" eq "Delete") {
    &get_lock;
$SPARTNO="$PARTNO";
&open_db;
delete $db{$SPARTNO};
open(logger,>> $LOG");
    print logger "$RUSER@$RHOST - modify $SUBM - $ESTABLISHMENT - $FDATE\n";
close logger;
&write_db;
&rel_lock;
}

sub get_lock{
    while( -f "${DATADIR}/.library.LCK" ){sleep 5}
    open(foo,> ${DATADIR}/.library.LCK")||die "Content-type: text/html\n\nCan
not get lock"; close foo;
};# end of get_lock

sub rel_lock{
    unlink "${DATADIR}/.library.LCK";
};# end of rel_lock

sub open_db{
    undef $db;
    open(foo,"${DATADIR}/retailstore.db")||die "Content-type: text/html\n\nCan not
open data file for read";
    while(<foo>){
        chop;
        ($STATE,$ESTABLISHMENT,$ADDRESS,$CITY,$PHONENUM,$LOC,$PARTNO)=split(/\
        /,$_,7);
        $db{$PARTNO}="$_";
    }
    close foo;
};# end of open_db

sub write_db{
    open(foo,> ${DATADIR}/retailstore.db")||die "Content-type: text/html\n\nCan
not open data file for write";
    foreach $key(sort(keys(%db))){
        print foo "$db{$key}\n";
    }
    close foo;
};# end of write_db

sub sort_by_partno{
    foreach $key(sort(keys(%db))){

```

```

    ($STATE,$ESTABLISHMENT,$ADDRESS,$CITY,$PHONENUM,$LOC,$PARTNO)=split(/\
     /,$db{$key});
    $sdb{$key}="$db{$key}";
}
};# end of sort_by_partno

sub sort_by_webloc{
    foreach $key(sort(keys(%db))){
        ($STATE,$ESTABLISHMENT,$ADDRESS,$CITY,$PHONENUM,$LOC,$PARTNO)=split(/\
         /,$db{$key});
        $I="$LOC$ESTABLISHMENT";
        $sdb{$I}="$db{$key}";
    }
};# end of sort_by_webloc

sub sort_by_city{
    foreach $key(sort(keys(%db))){
        ($STATE,$ESTABLISHMENT,$ADDRESS,$CITY,$PHONENUM,$LOC,$PARTNO)=split(/\
         /,$db{$key});
        $I="$CITY$STATE$PARTNO";
        $sdb{$I}="$db{$key}";
    }
};# end of sort_by_city

sub sort_by_address{
    foreach $key(sort(keys(%db))){
        ($STATE,$ESTABLISHMENT,$ADDRESS,$CITY,$PHONENUM,$LOC,$PARTNO)=split(/\
         /,$db{$key});
        $I="$ADDRESS$STATE$PARTNO";
        $sdb{$I}="$db{$key}";
    }
};# end of sort_by_address

sub sort_by_state{
    foreach $key(sort(keys(%db))){
        ($STATE,$ESTABLISHMENT,$ADDRESS,$CITY,$PHONENUM,$LOC,$PARTNO)=split(/\
         /,$db{$key});
        $I="$STATE$ADDRESS$PARTNO";
        $sdb{$I}="$db{$key}";
    }
};# end of sort_by_state

sub sort_by_establishment{
    foreach $key(sort(keys(%db))){
        ($STATE,$ESTABLISHMENT,$ADDRESS,$CITY,$PHONENUM,$LOC,$PARTNO)=split(/\
         /,$db{$key});
        $I="$ESTABLISHMENT$STATE$PARTNO";
        $sdb{$I}="$db{$key}";
    }
};# end of sort_by_establishment

&get_lock;
&open_db;
&rel_lock;
if( "$SORBY" eq "Establishment" ){&sort_by_establishment}
elsif( "$SORBY" eq "Address" ){&sort_by_address}
elsif( "$SORBY" eq "City" ){&sort_by_city}

```

```

elsif( "$SORTBY" eq "State" ){&sort_by_state}
elsif( "$SORTBY" eq "Web Address" ){&sort_by_webloc}
elsif( "$SORTBY" eq "Record No." ){&sort_by_partno};

print<<"EOA";
Content-type: text/html

<BODY BGCOLOR="#FFFFFF">
<CENTER><H1>Retail Store Library</H1></CENTER>
<HR>
<CENTER>
<FORM ACTION=/howie/cgi/library.cgi>
<TABLE BORDER="1">
<TR>
<FORM ACTION=/howie/cgi/library.cgi>

<TD><INPUT NAME="SUBMS" TYPE="SUBMIT" VALUE="Sort by">
<TD COLSPAN=7 ALIGN=LEFT>
<SELECT NAME="SORTBY"><OPTION SELECTED>$SORTBY
    <OPTION>Establishment
    <OPTION>Address
    <OPTION>City
    <OPTION>State
    <OPTION>Web Address
    <OPTION>Record No.
</SELECT>
</TR>

<TR>
    <TD>Establishment
    <TD>Address
    <TD>City
    <TD>State
    <TD>PhoneNum
    <TD>Record No.
    <TD>Web Location
    <TD>Add/Update/Delete
</TR>
</FORM>

<TR>
<FORM ACTION=/howie/cgi/library.cgi>
EOA

#print "<TD><SELECT NAME=\"ESTABLISHMENT\">\n";
#foreach $t(sort(@TYPES)) { print "<OPTION>$t\n" };
#print "</SELECT>\n";
#print "<TD><SELECT NAME=\"ADDRESS\">\n";
#foreach $t(sort(@CLASSES)) { print "<OPTION>$t\n" };
#print "</SELECT>\n";

print<<"EOB";
<TD><INPUT SIZE=10 NAME="ESTABLISHMENT">
<TD><INPUT SIZE=10 NAME="ADDRESS">
<TD><INPUT SIZE=10 NAME="CITY">
<TD><INPUT SIZE=10 NAME="STATE">

```

```

<TD><INPUT SIZE=10 NAME="PHONENUM">
<TD><INPUT SIZE=10 NAME="PARTNO">
<TD><INPUT SIZE=10 NAME="LOC">
EOB

#print "<TD><SELECT NAME=\"LOC\">\n";
#foreach $t(sort(@LOCS)) { print "<OPTION>$t\n" };
#print "</SELECT>\n";

print "<TD ALIGN=CENTER><INPUT NAME=\"SUBM\" TYPE=\"SUBMIT\""
VALUE=\"Add/Update\">\n</TR>\n";
print "</FORM>\n<FORM ACTION=/howie/cgi/library.cgi>\n";
print "<TR><TD COLSPAN=7 ALIGN=RIGHT>Enter Record Number to Remove from ";
print "the Database <INPUT SIZE=10 NAME=\"PARTNO\">\n";
print "<TD COLSPAN=4 ALIGN=CENTER><INPUT NAME=\"SUBM\" TYPE=\"SUBMIT\""
VALUE=\"Delete\">\n";
print "</FORM></TR>\n";
print "</TABLE></CENTER>\n";
print "<HR>\n";

open(library_out, ">-");
print "<FONT SIZE=-1><CENTER>\n<PRE>\n";
foreach $key(sort(keys(%sdb))) {
    ($STATE,$ESTABLISHMENT,$ADDRESS,$CITY,$PHONENUM,$LOC,$PARTNO)=split(/\
        /,$sdb{$key});
    write library_out;
}

<!----->

<!----->

print "</FONT></PRE></CENTER>\n";

print "</BODY>\n</HTML>\n";

format library_out_TOP =
Establishment          Address          City          State
PhoneNum      Web Address      Record Number
=====
=====

format library_out =
@<<<<<<<<<<<<< @<<<<<<<<<<<<< @<<<<<<<<<<
@<<<<<<<<< @<<<<<<<< @<<<<<<<< @>>>>>>>>
$ESTABLISHMENT, $ADDRESS, $CITY, $STATE, $PHONENUM, $LOC, $PARTNO
.

```

```

#!/usr/local/bin/perl
#
# perl script to track Retail Stores

# menu variables
@CLASSES= ('Network', 'BU', 'OS', 'Other', 'Cluster');
@LOCS= ('Mediabox', 'Shelf_1', 'Shelf_2', 'Shelf_3', 'Shelf_4');

# Create this on the fly.
#@STATELIST= ('Alabama', 'Alaska', 'Arizona', 'Arkansas', 'British Columbia',
#      'Northern California', 'Southern California', 'Canada', 'Colorado',
#      'Connecticut');

$SORBY="Sort by Establishment";
$SORTSTATE="Select From List";

# program data directories
$$DATADIR="/opt/HTTPS/WWW/staff/library";
$DATADIR="/www/htdocs/MANHATTAN/howie/cgi/library";
$$DATADIR="/www/htdocs/howie/cgi/library";
$LOG="$DATADIR/library.log";
$STAFF="/opt/HTTPS/WWW/staff/staff.txt";

# get remote user
open(foo,"finger \@${ENV{'REMOTE_HOST'}} |");
while(<foo>){ if(/(^S+).*(console).*/){$RUSER=$1} }
close foo;
$$RHOST=$ENV{'REMOTE_HOST'};

# set some default variables
chop($DATE=`/usr/bin/date +%m/%d/%y`);
chop($FDATE=`/usr/bin/date`);

# process QUERY_STRING
push(@INC,"/opt/HTTPS/WWW/cgi");
push(@INC,"/www/htdocs/MANHATTAN/howie/cgi");
#push(@INC,"/www/htdocs/howie/cgi");
require 'cgi-lib.pl';
&ReadParse(*in);
foreach $key(keys(%in)){
    if($key eq ESTABLISHMENT){$ESTABLISHMENT=$in{$key}}
    if($key eq ADDRESS){$ADDRESS=$in{$key}}
    if($key eq CITY){$CITY=$in{$key}}
    if($key eq PHONENUM){$PHONENUM=$in{$key}}
    if($key eq STATE){$STATE=$in{$key}}
    if($key eq LOC){$LOC=$in{$key}}
    if($key eq SORBY){$SORBY=$in{$key}}
    if($key eq SORTSTATE){$SORTSTATE=$in{$key}}
    if($key eq SUBM){$SUBM=$in{$key}}
    if($key eq SUBMS){$SUBMS=$in{$key}}
}
}

sub get_lock{
    while( -f "${DATADIR}/.library.LCK" ){sleep 5}
    open(foo,> "${DATADIR}/.library.LCK")||die "Content-type: text/html\n\nCannot
not get lock"; close foo;
};# end of get_lock

```

```

sub rel_lock{
    unlink "${DATADIR}/.library.LCK";
};# end of rel_lock

sub open_db{
    undef $db;
    open(foo,"${DATADIR}/retailstore.db")||die "Content-type: text/html\n\nCan not
open data file for read";
    while(<foo>){
        chop;
        ($STATE, $ESTABLISHMENT, $ADDRESS, $CITY, $PHONENUM, $LOC)=split(/ /,$_,7);
        $db{$CITY}="$_";
    }
    close foo;
};# end of open_db

sub write_db{
    open(foo,> ${DATADIR}/retailstore.db")||die "Content-type: text/html\n\nCan
not open data file for write";
    foreach $key(sort(keys(%db))){
        print foo "$db{$key}\n";
    }
    close foo;
};# end of write_db

sub sort_by_city{
    foreach $key(sort(keys(%db))){
        ($STATE,$ESTABLISHMENT,$ADDRESS,$CITY,$PHONENUM,$LOC)=split(/ /,$db{$key});
        $sdb{$key}="$db{$key}";
    }
};# end of sort_by_city

sub fill_state_pulldown{
    $statelistindex = 0;
    foreach $key(sort(keys(%db))){
        ($STATE,$ESTABLISHMENT,$ADDRESS,$CITY,$PHONENUM,$LOC)=split(/ /,$db{$key});
        $I="${STATE}${CITY}";
        $idb{$I}="$db{$key}";
    }

    foreach $key(sort(keys(%idb))){
        ($STATE,$ESTABLISHMENT,$ADDRESS,$CITY,$PHONENUM,$LOC)=split(/ /,$idb{$key});
        if ($statelistindex == 0) {
            @STATELIST[$statelistindex++] = $STATE;
        }

        if ($statelistindex > 0) {
            if ($STATE ne @STATELIST[$statelist-1]) {
                @STATELIST[$statelistindex++] = $STATE;
            }
        }
    }
};# end of fill_state_pulldown

sub sort_by_state{

```

```

foreach $key(sort(keys(%db))) {
    ($STATE,$ESTABLISHMENT,$ADDRESS,$CITY,$PHONENUM,$LOC)=split(/ /,$db{$key});
    $I="$${STATE}$$${CITY}";
    $sdb{$I}="$db{$key}";
    $$sdb{$key}="$db{$key}";
}
};# end of sort_by_state

sub sort_by_address{
    foreach $key(sort(keys(%db))) {
        ($STATE,$ESTABLISHMENT,$ADDRESS,$CITY,$PHONENUM,$LOC)=split(/ /,$db{$key});
        $I="$${ADDRESS}$$${CITY}";
        $sdb{$I}="$db{$key}";
    }
};# end of sort_by_address

sub sort_by_establishment{
    foreach $key(sort(keys(%db))) {
        ($STATE,$ESTABLISHMENT,$ADDRESS,$CITY,$PHONENUM,$LOC)=split(/ /,$db{$key});
        $I="$${ESTABLISHMENT}$$${CITY}";
        $sdb{$I}="$db{$key}";
    }
};# end of sort_by_establishment

#
# Start of real program...
#
&get_lock;
&open_db;
&rel_lock;

&fill_state_pulldown;

if( "$SORTBY" eq "Sort by Establishment" ){&sort_by_establishment}
elsif( "$SORTBY" eq "Sort by City" ){&sort_by_city}
elsif( "$SORTBY" eq "Sort by State" ){&sort_by_state}
elsif( "$SORTBY" eq "Sort by Address" ){&sort_by_address};

print<<"EOA";
Content-type: text/html

<HTML>
<BODY BGCOLOR="ffffff">
<CENTER><H1>Retail Stores</H1></CENTER>
<HR>
<CENTER>
<FORM ACTION="/howie/cgi/library-ro.cgi">

<TABLE BORDER="1">

<TR>
<FONT SIZE=+1>
Display all retail stores in a geographic region. <BR>

```


\$ESTABLISHMENT, \$ADDRESS, \$CITY, \$PHONENUM, \$LOC

(17)

Netmask Calculator

Network Number:	<input type="text" value="\$netn"/>
Existing Netmask:	<input type="text" value="\$fnetm"/>
OR	
Requested Networks:	<input type="text" value="\$nets"/>

GO **RE**

```
EOH if(("$netn" ne "")&&("$nets" ne "" || "$fnetm" ne "")){ &mainsub; } else{ print  
<<"USAGE";
```

Use the above form to verify an existing network and netmask or to calculate the appropriate network and netmask values for any network.

- Enter a network number (such as 205.185.225).
- Enter one of either:
 - The existing netmask.
 - The number of subnets needed.
- Then click on *GO*.

The program will round the number of subnets up to the next logical number of bits to be used. The bits used from the host address and added to the network address are displayed in **bold**.

```
USAGE } print <<"EOE";
```

Sun Internal Use Only!
Last modified: Apr 25, 1997
Created by Tom Southerland